Security and Provenance in M3GS for Cross-domain Information Sharing

Jingwei Huang and David Nicol
Information Trust Institute, University of Illinois at Urbana-Champaign
Email: {jingwei,dmnicol}@illinois.edu

***Abstract*—Modern military activities involve significant data sharing across security domains. We present the concepts and architecture of a Mission-oriented Multi-domain Multi-level security Graphics Server (M3GS) in the environment of GIG 2.0 and cloud computing. M3GS aims at providing information support for a dynamic team collaborating on a mission of warfighting, intelligence, anti-terrorism, or rescue and disaster relief; information providers input data (with various security labels in different security domains) into M3GS, and through M3GS, those data are displayed with proper widgets on the screens of information clients permitted to access; what data can flow to which screen is governed by security policies. While the Bell-LaPadula model is used to enforce traditional mandatory access control, a new challenge is that the data shared have different owners from different security domains, and are subject to their own security policies. We address this problem by using dynamic provenance-dependent attribute-based policies.**

I. Introduction

Modern military activities involve significant data sharing among diverse organizational entities (including different national entities). Each entity that shares information has its security policy—i.e., its own requirements on access and use of that data. The challenge becomes managing that sharing in a way that respects each entity's security policy.

Highly relevant, "need-to-share" [1], [14] is an emerging term reflecting the information management philosophy or culture, shifting from the traditional security management culture labeled with "need-to-protect" since 9/11. The background problem is that classified information with bureaucratic boundaries may be over protected or may not be provided by its custodians, so that some valuable information is not known and/or cannot be used by other need-to-use entities in duty of protecting nation's security. The management (or political) issue is balancing need-to-protect with need-to-share. A major research issue is managing information so as to meet both need-to-protect and need-to-share requirements.

Reflecting the needs of information sharing, the DoD is transforming its information infrastructure – the Global Information Grid (GIG) into the next generation (GIG 2.0) to enable the DoD vision of net-centric operations (NCO), which requires all information and services to be visible, accessible, understandable, and trusted across the DoD enterprise. In GIG 2.0, information, services, and applications will no longer be hosted and maintained by organizational and functional "stovepipe" systems; instead they will be resourced virtually on GIG Computing Infrastructure (CI) nodes spread across GIG's pooled resources [10]. GIG CI nodes range in scale from global enterprise CI nodes, regional CI nodes, to unit-level, modular, deployable CI nodes. This NCO vision can be achieved by using an assured cloud computing paradigm. We envision GIG 2.0 as a trustworthy enterprise cloud.

In this paper, we develop the concepts and sketch the architecture of a Mission-oriented Multi-domain Multi-level security Graphics Server (M3GS) for cross domain information sharing in the environment of GIG 2.0 or DoD clouds. M3GS could be composed of a set of cloud services, or stand alone, as needed. It aims at providing information support for a dynamic team collaborating on a mission of warfighting, intelligence, or rescue and disaster relief. Information providers input data (typically classified) into M3GS; through M3GS, data are displayed with proper widgets on the screens of information clients who need that data to complete the assigned tasks in the mission. Security policies govern the flow of data to screens.

A highly relevant concrete example is a graphics server on the ground, which displays information on the screens of the cockpits of aircrafts. Context-sensitive constraints exist with respect to the display's viewers; e.g., sensitive map and targeting data ought not be displayed when the plane is on the ground being serviced by mechanics. Other constraints might govern what data is permitted to be overlaid on a given display, or what resolution of map is permitted to be displayed depending on the security level of the viewer. Security-sensitive sharing among entities requires interoperation between different nations' multi-level security labeling—is it permissible for data with a "secret" U.S. label to be viewed by a German observer with "VS-Vertraulich" clearance? The server must be able to handle this level of cross-domain policy resolution. Furthermore, some policies may depend on the provenance of the input and not just security labels. For instance, in an international coalition of ships serving to interdict Somali pirates, a ship from the United States Navy may receive reports from ships of other nations, and generate intelligence maps for them all. The USN graphics server may be constrained by Saudi Arabian policy from directly including any Saudi data on a display generated for an Israeli ship. The policies resolved by the server may thus depend on data provenance as well.

This material is based upon research sponsored by Rockwell Collins Inc., and by the U.S. Air Force Research Laboratory and the U.S. Air Force Office of Scientific Research, under agreement number FA8750-11-2-0084. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

978-1-4673-3/12/$31.00 ©2013 IEEE

The content of this paper is as follows: section II presents the concepts and an architecture for M3GS; section III discusses the security policies for M3GS; section IV further discuss provenance-dependent policies and presents a use case illustrating the concepts; finally in section V, we summarize the work and discuss further research.

II. Concepts and Architecture

The following entities and concepts play important roles in our conception of M3GS:

- *Screen*: a graphical frame in which information is displayed by *widgets*.
- *Message*: the basic information unit going through graphics server. A message may be in the form of picture, text, video stream, radar map, and so forth. Each message is displayed with a widget, and is marked up with metadata about security labels, data usage rules, and provenance.
- *Widget*: a piece of software that accepts data as input and places graphical output within the display subframe allocated to it.
- *Information client*: a human observer of one or more screens, who uses received information for the completion of an assigned mission.
- *Information manager*: a human who (statically) associates data to each widget, and (statically) assigns widgets to particular screens.
- *Information provider*: an entity that provides input to the graphics server.

To make these notions concrete, consider the cockpit of an aircraft. The control console may be partitioned into a number of different screens, and widgets defined to display information such as bearing, fuel levels, maps, etc. In flight the pilots would be information clients (but on the ground a mechanic might be an information client); information producers could be sensors, video streams from UAVs, text-based messages from command, maps, and so on. Another typical example is a big screen in a command center; various information from diverse security domains is integrated and displayed in one or more screens, and some information (after processing or not) is sent to remote screens of information users who working in the front-line of a mission.

Every information client, manager, and provider has a security label in a security domain, reflecting one's security clearance (if one has); all devices have security labels (to be discussed later); all the data originate from diverse security domains, and carry meta-data describing their (origin) domain-specific security labeling and provenance. This meta-data is a key element of the policy resolution problems we address. Figure 1 gives a high level view of the architecture. The architecture consists of *network portals, policy enforcement Point (PEP), databases, policy engine,* and the *graphics engine*. All team members communicate with MMGS through the classified networks of GIG, such as NIPRNet (for "unclassified" level), SIPRNet (for "secret" level), and JWICS (for "top secret" level). MMGS's GIG portal conducts authentication of all accessors to MMGS; when the Policy Enforcement Point (PEP) receives an access request, it submits the request, (subjectId, resourceId, access type, context), to the Policy Engine, which identifies and configures proper policies to govern the requested resource, collects (locally in MMGS' data repositories and/or globally from GIG) the attributes of both resource, subject, and other context information (e.g. mission statement, assurance levels of access channel and devices), makes a decision on that access request, and returns the decision to PEP; if granted, PEP executes the requested access (e.g., update a message, screen, or some metadata.)

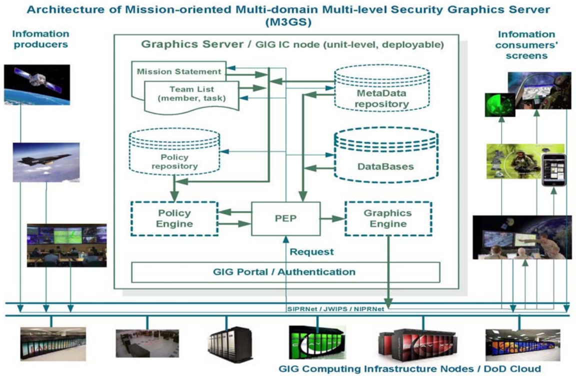
Architecture of Mission-oriented Multi-domain Multi-level Security Graphics Server (M3GS)
Infomation producers
Graphics Server / GIG IC node (unit-level, deployable)
Infomation consumers' screens
Mission Statement
Team List (member, task)
MetaData repository
Policy repository
DataBases
Policy Engine
PEP
Graphics Engine
Request
GIG Portal / Authentication
SIPRNet / JWIPS / NIPRNet
GIG Computing Infrastructure Nodes / DoD Cloud

Fig. 1. An architecture of M3GS

Security for a graphics server has several aspects. The most obvious is that the platform and the communication be secured from outside interference. Cryptography, specialized hardware, hardened communication protocols all contribute to this sort of security. While important, this is not the focus of our work. We assume that the computing hardware (for stand-alone M3GS) rests on a trusted computing base protected by some hardware assistance such as a Trusted Computing Module. For our purposes we also assume that all communication with the graphics server is secure. A second aspect related to security addresses the potential that the system permit an insider to obtain information through covert means, taking advantage of the fact that data from diverse security domains with diverse security labels come together in the graphics server. The very architecture of the graphics server thwarts this kind of insider attack. In our earlier work on the high assurance graphics engine [18], a user does not have a view into the frame buffer where the screen output is assembled. A third aspect concerns elimination of potential information leak in the information flow of a M3GS, i.e. it is about what data are allowed and how they are displayed by widgets on which screens. This too is critically important, and is our major focus in this paper. There is not a clear division between "security" and "safety" in this realm. For our purposes we suppose that appropriate display policies reflecting safety-critical requirements can be developed, expressed, and integrated with security policies within the graphics server's policy database. We focus on secured information flow management issue.

Security becomes even more complex, when a M3GS is implemented with cloud computing. Similar to [21], a basic

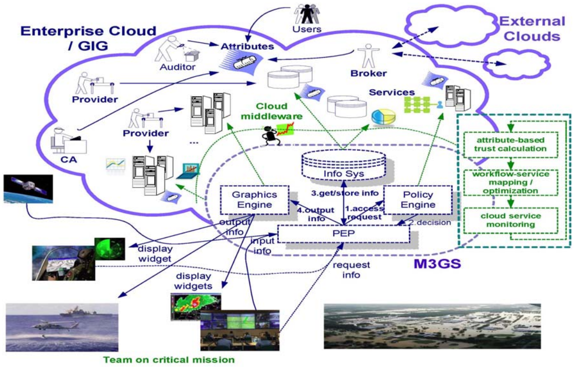

Fig. 2. M3GS deployed over the clouds.

issue is how to map M3GS components over cloud services with required security labels, i.e. to make the map satisfy Bell-LaPadula model [13]. Consider that in GIG all users, information, and computing resources have security labels. Cloud workflow access control is expected in GIG. We are currently developing access control models for assured cloud computing, by combining lattice-based access control (such as BLP model), attribute-based policies, and role-based access control model [12]. A further issue is optimization of the M3GS workflow mapping over clouds to maximize the performance and security of M3GS, based on the attributes of cloud services. A high level abstraction is given in figure 2, detailed discussion will be given in another paper.

Complexity abounds in this system. In this initial effort we try to manage that complexity by identifying a logical framework focusing on secured cross-domain information flow management on M3GS. The richness of the attributes on which decisions are made about what may be displayed, how it may be displayed, and to whom it may be displayed requires us to give considerable thought to suitable policies governing the information flow through M3GS.

III. Security Policies

In the context of military and intelligence information sharing, policies of interest naturally include US government classification system. With some small variations, most nations including US follow the British model of a 5 level classification : *Top Secret*, *Secret*, *Confidential*, *Restricted*, and *Unclassified*. NATO nations have already worked though a joint information sharing classification with 5 (slightly different) levels : *COSMIC TOP SECRET*, *FOCAL TOP SECRET*, *NATO SECRET*, *NATO CONFIDENTIAL*, and *NATO RESTRICTED*.

Such systems are used for both *classification of information* and *security clearance* of individuals, assigning a *security label* (also called *security level*) to a piece of information or a container of information and to an individual separately. A *security label* consists of (1) a *sensitivity level*, and (2) a set of *need-to-know categories*. A *sensitivity level* is evaluated based upon the damage to the defining organization's security caused by the information leakage. *Need-to-know categories* are categories of specific job-related need-to-know information, or compartment of sensitive information related to specific subject areas or national-security topics or programs. To access a classified object, an individual must have sufficient security clearance.

In computer systems, the above management system is represented mathematically as the Bell-LaPadula (BLP) model [13]. Briefly, BLP model expresses "no read up" and "no write down" principles. BLP is focused on preventing leakage of information at high security sensitivity levels to lower sensitivity levels and leakage of information outside of need-to-know categories. The Biba model [2] is BLP's dual, focusing on preserving the integrity of information. Information flow model [7] focuses on preventing unauthorized information flow from one compartment to another. The rules defined by those formal models provide the general framework of policies we consider.

With respect to sharing of classified data cross security domains, countries that cooperate have largely already worked through the mapping of security levels, or definition of security levels based on the organization within which they cooperate (e.g. NATO-X classifications). Comparison of security labels from such different domains is a topic known in the literature as "secure interoperation", e.g. [3], [6], [19]. Basically, in each classification system, there is a partial order relation among security labels (pairs of sensitive level and need-to know category set); the relation forms a "lattice", which constrains access control in a security domain, so called "lattice-based access control". Secure interoperation requires to merge multiple lattices into one, by building a mapping among sensitive levels in different domains and a mapping between need-to-know categories. The mapping must be consistent; that is to say, no contradiction introduced; and in any single domain, the partial order relation remains unchanged after introducing the mapping.

The above discussed lattice-based access control models (BLP formalisms as a representative) capture very well the hierarchical structure of a security classification system; however, non-hierarchical classification schemes also exist. For example, in US government classification system, sensitive data are also marked with *code words* representing need-to-know categories in SCI/SAP, and *caveats* for the releasability of sensitive data, like "ORCON", "NOFORN", and "REL TO" [8], [20]; non-hierarchical policies are based on caveats that define the conditions under which the marked information may be released. In our very context of data sharing among security domains through M3GS, more distinguishing features need to be considered in the security policies governing the sharing.

First, military and intelligence information sharing across security domains is generally sensitive to provenance. It is easy to find cases where sensitive data in a security domain is prohibited from leaking to some other specific domains, as the earlier example of an international pirate interdiction fleet points out. Provenance-dependent policy addresses such issues. Detailed discussion is given later in sections IV.

Secondly, in addition to provenance, policies may need to consider other attributes regarding the information, the requester, and the context, particularly, the information usage context such as the specific use of the information, where the information should be used, and the time-frame of its use. A custodian of secure information is usually seriously concerned about such information usage-context.

Finally, information-specific rules are the rules associated with specific shared data and for special cases or exceptional cases to consider. Some special information may need special protection or special sharing procedures; some information may be the exceptions of general policies. Some special rules enhance security constraints, and some others may relax security constraints.

Unlike the Bell-LaPadula model whose policy is known a priori, provenance-dependent policy, context-dependent policy, and information-specific rules are defined by each security domain, and are not known in advance. We may assume that provenance-dependent policy and context-dependent policy will be given by each security domain when the graphics server is configured, but these policies may later need revision while the system is in use; information-specific rules are attached to incoming data, i.e., policy may be part of dynamic run-time knowledge (in parallel to run-time data), rather than pre-defined static knowledge. One of our challenges is to handle dynamically changing policy, in real-time.

We view a policy as a set of rules, where each rule can be expressed as a logical formula in First Order Logic (FOL), describing the attributes of a requester, the attributes of the requested information, and the context of access. This broad view is supported by the emerging XACML standard [16]. XACML is declarative, a policy is a set of rules that specify access decisions as a function of attributes associated with a "Subject", "Resource", "Action", and "Environment". This attribute-based approach enables us to represent various security policies discussed earlier in a unified form. Both the security labels (including sensitivity level, and need-to-know categories) of information items (as well as facilities), and the security clearance of individuals are treated as attributes; provenance of information and context of information usage are also treated as attributes. Different from the standard attribute-based models where rules are given based on the values of the attributes of involved entities, provenance involves tracing a "provenance graph", so that provenance based policy are more complex. We discuss more on provenance in § IV.

IV. Provenance Modeling and Management

We have already alluded to the possibility that policy may refer to the origins of data, i.e., data provenance. According to the Oxford English Dictionary, provenance is the origin and derivation of an object. Evolving from fine art, where provenance is the documented records about the history of a work of art, provenance is, generally speaking, the information about the creation, derivation, and/or modification of an object, and other historical information about that object, such as chain of custody, context information. Interest in provenance has grown in recently years within computer science [4], [5], [11], [15], particularly in the fields of scientific workflow, database, AI, and security.

Provenance is an important basis for judging the authenticity, integrity, quality, validity, and value of information; for validating and repeating scientific experiments; for evaluating workflow; for analyzing reasons of an operation failure; for conducting audit; for declaring credit and responsibility; and for enforcing legitimate use.

Provenance matters in cross-domain information sharing for the following reasons:

- information clients are concerned about the authenticity, integrity, quality, validity, and value of information, attributes critical for decisions on how to use it;
- information producers and owners are concerned about where their information goes, who has access to it, the purposes for which it is used, and protection of their information confidentiality;
- information managers need to know the provenance of information, in order to manage information flow to enforce each domains security policies;
- an information server may need to document information processing, for purposes of validation, failure analysis, and audit.

Given the importance of provenance, the next question asks how we can represent and handle it. DoD has developed DDMS (DoD Discovery Metadata Specification) [9] as DoD's metadata standard for secure data sharing, which includes IC ISM (Intelligence Community Information Security Markings) [17]. The metadata categories defined in DDMS cover security and provenance, e.g. classification (sensitivity level only), creator, publisher, contributor, date of creating, posting, and expiration, subject-coverage, geospatial coverage, temporal coverage, related resources and relationships. It appears then that DDMS can be used as the standard of provenance metadata for marking up messages going through the graphics server.

To illustrate the role of data provenance with respect to policy in an M3GS, we describe a simple use-case that exercises several aspects of the policy architecture. The use case shows how information is gathered and passes through several steps before being presented to the M3GS, and that display of the information depends in part on its provenance, and also requires resolution of cross-domain security labels. The example uses a number of data provenance attributes with terms that are intended to be descriptive (e.g., "createdBy", "createdOn", "dependentOn") rather than direct reference to the DDMS standard. We believe the mapping to DDMS description is straightforward.

We imagine a scenario in Afghanistan where US forces are training Afghanistan army units in counter-insurgency. US Forces there host an M3GS that creates graphics for both US and Afghani commands. Suppose now that CIA station stn-1 receives, from an informer inside a terrorist group, a message that the insurgent group plots to attack Afghan army facility F. The CIA station annotates this information as follows:

Info-1: Insurgents plan to attack facility F;
createdBy: CIA;
createdOn: 2010-04-23-21:30 (yyyy-mm-dd-hh:mm);
dependentOn: Info-2;
securityLevel: (top secret, {Insurgency, Facility-F}) [2010-04-23, 2010-04-25]; (secret, {Insurgency, Facility-F}) [2010-04-26, -].

Info-2: Insurgent group HIG plots to attack facility F;
createdBy: CIA-stn-1;
createdOn: 2010-04-23-21:25;
dependentOn: Info-3;
securityLevel: (top secret, {Intelligence, Insurgency, Afghanistan});
specialAccessPolicy: policy-2.

Policy-2: Info-3 can only be read by CIA head-of-station at stn-1.

Some of the security labels above are annotated with ranges of time during which the associated label applies. This makes security classification time-dependent, which for instance prohibits revelation of the information to lower levels of clearance while the informer makes an escape.

Info-3 is the message provided by informer; this message and its provenance have top-secret classification, with an additional constraint of case-based evaluation by the CIA head-of-station at stn-1.

Through the server, based on Info-1, US Forces headquarters sends a UAV to monitor the region around facility F. Later, the observer of the UAV video feed finds that a suspect truck is driving towards facility F. The observer creates Info-4 :

Info-4: Video stream of a suspect truck approaching facility F;
securityLevel: (confidential, {Insurgency, Facility-F});
{createdBy: UAV-F;
createdOn: 2010-04-25-12:30pm;
securityLevel: (secret, {Insurgency, UAV, Facility-F})};

Note that Info-4 is a source, it does not depend on anything else. In a data provenance graph it would be a root. Its provenance meta-data is contained within this statement inside of the curly braces, and has its own security label, which in this case is higher than that of the data itself. This reflects the need to make data available for use, but protect its source.

Based on Info-1 and Info-4, US Forces headquarters creates two data artifacts. One is a map with a "current position" marker of the truck; this is Info-5.

Info-5: Map showing position of truck approaching Facility-F with intent to attack.
createdBy: USF-AFG;
createdOn: 2010-04-25-12:31pm;
dependentOn: Info-1, Info-4;
securityLevel: (confidential, {Insurgency, Facility-F});

Another data artifact is created, a streaming video from the UAV tracking the truck. The security sensitivity is the same, but UAV is added as a need-to-know category.

Info-6: Streaming video of truck approaching Facility-F with intent to attack.
createdBy: USF-AFG;
createdOn: 2010-04-25-12:31pm;
dependentOn: Info-1, Info-4;
securityLevel: (confidential, {Insurgency, UAV, Facility-F});

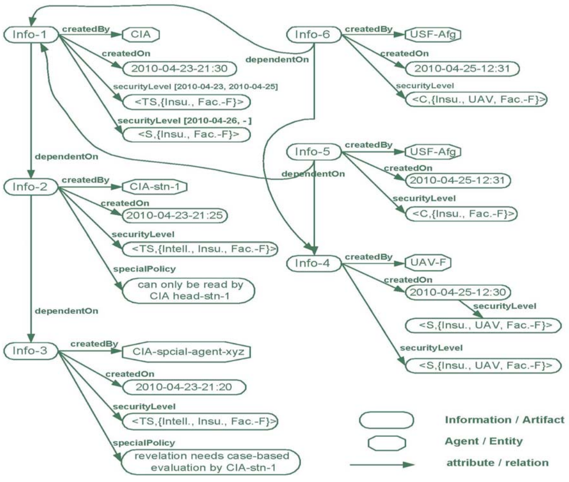

Fig. 3. Provenance Graph for the Example

Figure 3 illustrates a data provenance graph that documents Info-5, Info-6 and their provenance.

The assembly of meta-data such as illustrated by Figure 3 is assumed to occur outside of the M3GS architecture, but in a form that the M3GS is able to parse. Meta-data in the provenance graph may also have certain security labels and set of special access policies. We also assume that the meta-data may contain policy statements about use of the data. Examples are seen in Figure 3 at Info-2 and Info-3, nodes pointed to with "specialPolicy" arcs. We may imagine that other policy statements might be included as part of the meta-data, expressed e.g. in XACML. As the policy accompanies the data, we refer to this as *dynamic* policy, which, together with *static* policy within the M3GS policy database (e.g., BLP write rules), governs generation of the screens.

Info-5 and Info-6 are mapped as input to two different widgets, but these widgets are both mapped to two different screens. For each screen, policy governs whether these widgets are allowed to update it.

If we consider an instance of a widget as an active subject that writes to screens, then we must assign a security label to it. The natural definition is that the widget's security sensitivity is the maximum sensitivity among all of the inputs it acts upon to produce output, and that its set of categories is the union of the categories of all its inputs. This definition preserves the "no write-down" aspect of BLP, and ensures all the possible need-to-know categories that a write might entail are represented. A screen's security label is that of its observer; if there are multiple observers, then its sensitivity level ought to be the minimum among all its observers, and its need-to-know category the intersection of all its observers.

In the example, one screen is viewed by US command, the other by Afghani command. We may suppose that Afghani security sensitivity labels have been aligned with the US system, but that the two systems have different categories. Subset relationships have been established though. The US *Facility-F* category is subsumed by an Afghani regional category, and US and Afghani categories of *Insurgency* are considered to be identical. The security sensitivity for the US command is *Secret*, and its set of categories include *Insurgency*, *UAV*, and *Facility-F*. The security sensitivity level for the Afghani command is *Confidential*; its categories include the regional one that subsumes *Facility-F*, and *Insurgency*. It does not however include the *UAV* category.

Consider first the policy applied to the US command screen. Security labels for Info-5 and Info-6 allow widgets for both to write to that screen. Not only that, the widgets have access to all information artifacts in their provenance graphs which also have security labels that allow them to be written to the screen. The only such artifact is Info-4 though, as the sensitivity levels of Info-1, Info-2, and Info-3 are all *Top Secret*. Info-4 might identify, for instance, the particular UAV in flight and contact information of its remote operator.

Unlike the US command screen, the Afghani command screen may not be updated by a widget instance fed by Info-6—lacking a *UAV* category, the BLP write rule prohibits the update. This simple mechanism allows the US to protect the source of information about the moving truck, while the Info-5 widget (map with truck marker) provides essential information about that vehicle.

V. Summary and Further Work

In this paper, we identify the needs, develop the concepts, and sketch an architecture for a mission-oriented multi-domain multi-level security graphics server (M3GS) as a type of tools supporting cross-domain information sharing, particularly, providing secure information support for a dynamical team of members from different security domains collaborating on a mission; the server produces the contents of screens fusing information from a variety of sources in different security domains — with a variety of security labels. Policies govern what data can flow to which users' screens; and our principal interests focus on those security policies and their expression. In particular, we identify attribute-based approach as a general framework to express the general policies from traditional military access control hierarchical schemas modeled by lattice-based models such as BLP, the non-hierarchical classification schemas such as caveats, information-specific and context-dependent policies governing information usage, and the provenance-dependent policies. We have implemented a prototype to demonstrate the concepts of M3GS. Corresponding to the architecture of M3GS, the prototype consists of four modules: PEP, policy engine (PDP), databases, and graphics engine. We assume that all data are annotated with xml metadata with respect to security labels, provenance, and policy Ids pointing to applied special policies. Our policy engine is based on the model of XACML, and implemented with Java.

Follow-on work will refine the ideas broadly sketched here. In particular, we will develop ontologies to precisely specify the attribute-based policy model in both logic and language levels, to facilitate metadata annotation, policy specification, and policy decision on M3GS; we will develop M3GS cloud workflow access control model, and trust-based cloud workflow optimization to maximize the trustworthiness of M3GS.

Acknowledgment

We would like to thank Raymond J. Richards and Yves LaCerte in Rockwell Collins Inc. for their valuable discussion and insight.

References

[1] AFCEA-Intelligence-Committee. The need to share: The u.s. intelligence community and law enforcement, 2007. http://www.afcea.org/mission/intel/documents/SpringIntel07whitepaper_000.pdf.

[2] K. J. Biba. Integrity considerations for secure computer systems. Technical report, April 1977. MITRE Technical Report 3153.

[3] P. A. Bonatti, M. Sapino, and V. S. Subrahmanian. Merging heterogeneous security orderings. In *Journal of Computer Security*, pages 25–27. Springer-Verlag, 1996.

[4] J. Cheney, P. Buneman, and B. Ludäscher. Report on the principles of provenance workshop. *SIGMOD Rec.*, 37(1):62–65, 2008.

[5] S. B. Davidson and J. Freire. Provenance and scientific workflows: challenges and opportunities. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1345–1350, New York, NY, USA, 2008. ACM.

[6] S. Dawson, S. Qian, and P. Samarati. Providing security and interoperation of heterogeneous systems. *Distributed and Parallel Databases*, 8(1):119–145, January 2000.

[7] D. E. Denning. A lattice model of secure information flow. *Commun. ACM*, 19(5):236–243, 1976.

[8] Director of Central Intelligence Directive 1/7. Security Controls on the Dissemination of Intelligence Information, June 1998. http://www.fas.org/irp/offdocs/dcid1-7.html.

[9] DoD. DoD Discovery Metadata Specification (DDMS) Version 3.0, January 2010. http://metadata.dod.mil/mdr/irs/DDMS/.

[10] DoD CIO. Department of Defense Information Enterprise Architecture, Version 1.2, May 2010. http://cio-nii.defense.gov/sites/diea/products/DoD_IEA_v1_2_7_May_2010.pdf.

[11] J. Huang. *Knowledge Provenance: An Approach to Modeling and Maintaining The Evolution and Validity of Knowledge*. Ph.D. Thesis, University of Toronto, http://hdl.handle.net/1807/11112, Dec. 2007.

[12] J. Huang, D. M. Nicol, R. Bobba, and J. H. Huh. A framework integrating attribute-based policies into RBAC. SACMAT '12, 2012. http://dl.acm.org/citation.cfm?id=2295136.2295170.

[13] L. La Padula and D. E. Bell. Secure computer systems: A mathematical model. Technical report, 1973. MITRE Technical Report 2547, Vol. II.

[14] Markle-Foundation. Mobilizing information to prevent terrorism – accelerating development of trusted information sharing environment, 2006. http://www.markle.org/downloadable_assets/2006_nstf_report3.pdf.

[15] L. Moreau, P. Groth, S. Miles, J. Vazquez-Salceda, J. Ibbotson, S. Jiang, S. Munroe, O. Rana, A. Schreiber, V. Tan, and L. Varga. The provenance of electronic data. *Commun. ACM*, 51(4):52–58, 2008.

[16] OASIS. eXtensible Access Control Markup Language (XACML) version 2.0, OASIS standard, 2005.

[17] Office of the director of national intelligence. Ics2007-500-2 intelligence community standard for information security marking metadata, 2007. http://metadata.dod.mil/mdr/irs/DDMS/documents/ICS2007-500-2SecurityMarkingMetadata.pdf.

[18] H. Okhravi and D. M. Nicol. Trustgraph: Trusted graphics subsystem for high assurance systems. In *Proceedings of IEEE Annual Computer Security Applications Conference (ACSAC'09)*, December 2009.

[19] B. Shafiq, J. B. Joshi, E. Bertino, and A. Ghafoor. Secure interoperation in a multidomain environment employing rbac policies. *IEEE Trans. on Knowledge and Data Engineering*, 17(11):1557–1577, November 2005.

[20] The White House. Executive order 13526 - classified national security information, Dec. 2009. http://www.whitehouse.gov/the-press-office/executive-order-classified-national-security-information.

[21] P. Watson. A multi-level security model for partitioningworkflows over federated clouds. In *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, pages 180 –188, 29 2011-dec. 1 2011.